Univerzitet u Nišu

Elektronski fakultet

Steganografija

Steganografija slike uz pomoć Chaos Based Algorithm

Seminarski rad

Predmet: Digitalna forenzika

Student:

Tamara Milovanović, 1647

Mentor:

Prof. Dr. Bratislav Predić

UVOD

Steganografija je umetnost i nauka skrivanja informacija unutar drugih naizgled primitivnih medijuma poput slika, audio datoteka, video datoteka ili čak teksta, na način da prisustvo skrivenih informacija nije lako otkriti. Osnovni cilj steganografije je obezbediti prikrivenu komunikaciju, gde je prisustvo tajne poruke sakriveno, a ne njen sadržaj. Pitanje koje se često postavlja je da li je stagnografija deo kriptografije? Odgovor je ne. Steganografija se često smatra blisko povezanom sa kriptografijom, ali su njihovi ciljevi i pristupi različiti. Iako oba polja imaju za cilj zaštitu komunikacije, postoji bitna razlika između njih. Kriptografija se bavi šifriranjem informacija kako bi se osigurala tajnost komunikacije tako da samo odabrani primalac može da ih dešifruje. Otvoreno je da postoji komunikacija, ali je sadržaj zaštićen od neovlašćenog pristupa. Steganografija, s druge strane, ne šifrira informacije, već ih krije unutar drugih medijuma na način koji ne privlači sumnju. Cilj steganografije nije da spreči otkrivanje prisustva komunikacije, već da sakrije samu komunikaciju. Na taj način, steganografska poruka može biti prisutna, ali neprimetna. Dakle, iako su kriptotgrafija i steganografija oba dela šireg polja informacione sigurnosti, one imaju različite svrhe i metofe.

Ideja da steganografija ima biološke ili fiziološke korene je intrigantna. Primeri onoga što danas prepoznajemo kao steganografiju zaista se mogu naći u životinjskom carstvu. Na primer, kada psi pokušaju da sakriju nešto što su uradili i budu uhvaćeni, mogu pokazati ponašanja koja podsećaju na stid. Slično tome, među šimpanzama ili vukovima, mužjaci nižeg ranga mogu pokušati da se pare sa ženkama koje pripadaju mužjacima višeg ranga na prikriven način.

Ovi primeri ilustruju da tajni i prikriveni načini komunikacije i delovanja postoje među životinjama, baš kao što postoje i u ljudskom društvu. U ljudskim interakcijama, deca mogu prenositi poruke u školi kako bi izbegla otkrivanje od strane nastavnika, dok pojedinci koji se bave kriminalnim aktivnostima mogu smišljati planove u tajnosti kako bi izbegli otkrivanje od strane zakonodavstva

Steganografija prevazilazi kriptografiju u tome što ne krije samo sadržaj već i samu postojanje tajne poruke. Koncept skrivanja informacija pred očima, kako se može primetiti i u životinjskom carstvu i u ljudskom društvu, sugeriše da steganografija ima duboke korene u našim biološkim i društvenim ponašanjima. Postavlja se pitanje gde se sve može sakriti poruka. U digitalnom svetu to radi u slikama, audio datoteke, video datoteke, tekstualne datoteke.

U audio datotekama, poruka se može sakriti modifikovanjem frekvencija, amplituda ili vremenskih intervala zvukova.

Poruka se može sakriti u video datotekama putem sličnih tehnika koje se koriste za slike i audio datoteke, kao i modifikovanjem pokreta ili boja.

Tekstualna poruka se može sakriti unutar drugog teksta, koristeći na primer specifične rasporede slova ili korišćenje posebnih fontova ili karakteristika teksta.

Steganografija slika je tehnika skrivanja poruka unutar digitalnih slika. Postoje različiti pristupi za implementaciju ove tehnike, ali jedan od najčešćih metoda je modifikacija najmanjih bitova piksela u slici. Ovi mali, gotovo neprimetni, izmene ne narušavaju vizuelni izgled slike, ali mogu sadržavati skrivenu poruku.

Postoji nekoliko tehnika koje se koriste za steganografiju slika:

1. **LSB (Least Significant Bit) metoda**: Ovo je jedna od najjednostavnijih metoda steganografije slika. Koristi se modifikacija najmanjih bitova (obično najmanje značajnih) piksela u digitalnoj slici kako bi se prenela poruka. Ove izmene su obično tako male da nisu vidljive ljudskom oku.
2. **Maskiranje boja**: U ovoj tehnici, određeni delovi slike se mogu malo izmeniti tako da tačke boja budu promenjene za vrlo malu vrednost. Ove promene su dovoljno male da ne budu primetne, ali mogu sadržavati skrivenu poruku.
3. **Frekvencijsko skrivanje:** Ova tehnika se fokusira na modifikaciju frekvencija u Fourierovom spektru slike, što rezultira promenama u originalnoj slici. Ove promene mogu biti veoma suptilne i teže su primetne od direktnih izmena piksela.
4. **Korišćenje skrivenih informacija u formatu slike:** Neki formati slika, poput JPEG-a, mogu sadržavati "prazne" delove koji se ne koriste za prikazivanje slike. Ovi delovi se mogu koristiti za skrivanje dodatnih informacija bez uticaja na kvalitet slike.
5. **Chaos-based algoritam:** Steganografija slika sa algoritmima baziranim na haosu koristi princip haotičnih sistema kako bi sakrila poruke unutar digitalnih slika. Ovi algoritmi koriste haotične dinamičke procese kako bi stvorili nelinearne transformacije slike, čime se omogućava efikasno skrivanje informacija.

U ovom radu detaljnije ce biti obrađeni i implementiran Chaos-based algoritam.

Teorija haosa je grana matematike koja proučava kompleksne sisteme koji su izuzetno osetljivi na početne uslove, što dovodi do nepredvidivog ponašanja poznatog kao haos. Ovi sistemi često pokazuju nelinearne dinamike, što znači da male promene u početnim uslovima mogu dovesti do drastično različitih rezultata tokom vremena.

U kontekstu steganografije, teorija haosa je relevantna zbog svoje sposobnosti generisanja prividno nasumičnih i nepredvidljivih uzoraka. Teorija haosa pruža tehnike za stvaranje složenih i prividno nasumičnih transformacija pokrivajućeg medijuma, što otežava neovlašćenim osobama da otkriju prisustvo skrivenih informacija.

Algoritmi bazirani na haosu mogu se koristiti za generisanje ključeva za enkripciju, odabir lokacija unutar pokrivajućeg medijuma za ugrađivanje tajne poruke, pa čak i enkripciju same poruke. Nepredvidiva priroda teorije haosa poboljšava

sigurnost steganografskih tehnika, čineći izuzetno teškim za otkrivanje ili dešifrovanje skrivenih informacije bez poznavanja specifičnih haotičnih parametara korišćenih u procesu kodiranja.

Ukratko, teorija haosa pruža osnovu za razvoj pouzdanih i sigurnih steganografskih metoda korišćenjem kompleksnog i nepredvidljivog ponašanja haotičnih sistema kako bi se informacije efikasno sakrile unutar pokrivajućih medija.

Teorija haosa

Teorija haosa je grana matematike i fizike koja proučava kompleksne sisteme koji pokazuju izuzetno osetljivu zavisnost od početnih uslova. Ovi sistemi su deterministički, što znači da njihovo buduće ponašanje potpuno određuje početno stanje i pravila koja upravljaju njihovim razvojem. Ipak, oni mogu pokazivati prividno nasumično i nepredvidivo ponašanje tokom vremena. Teorija haosa se bavi razumevanjem i karakterizacijom dinamike takvih sistema.

Pregled nekoliko ključnih definicija i svojstva teorije haosa

1. **Deterministički sistem**: Teorija haosa se bavi determinističkim sistemima, što znači da njihovo buduće stanje potpuno određuje početno stanje i pravila koja upravljaju njihovim evolucijom. Iako su deterministički, haotični sistemi mogu pokazivati izuzetno kompleksno i nepredvidivo ponašanje.

2. **Osetljiva zavisnost od početnih uslova:** Ovo se često naziva "efekat leptira", gde male razlike u početnim uslovima mogu dovesti do potpuno različitih rezultata tokom vremena. Čak i male varijacije u početnom stanju haotičnog sistema mogu rezultirati značajnim odstupanjima u njegovoj trajektoriji.

3. **Nelinearnost**: Haotični sistemi su obično nelinearni, što znači da njihovo ponašanje nije proporcionalno njihovim ulazima. Umesto toga, interakcije između različitih komponenti sistema mogu dovesti do složenih i često nelinearnih odnosa, doprinoseći pojavi haotičnog ponašanja.

4. **Čudni atraktori**: Haotični sistemi često pokazuju ponašanje koje konvergira ka podskupu svog prostora stanja poznatom kao čudni atraktor. Čudni atraktori imaju fraktalnu strukturu i predstavljaju dugoročno ponašanje sistema, hvatajući njegovu složenu dinamiku i ponavljajuće obrasce.

5. **Fraktalna geometrija**: Fraktali su geometrijski objekti koji pokazuju samopodudarnost na različitim skalama. Oni su često povezani sa haotičnim sistemima zbog svoje sposobnosti da opišu kompleksne i nepravilne oblike koji nastaju u haotičnoj dinamici.

6. **Periodičnost unutar haosa**: Iako haotični sistemi izgledaju nasumično, oni još uvek mogu pokazivati periodično ponašanje ili sadržavati periodične orbite ugrađene u svoje haotične trajektorije. Ovi periodični elementi mogu biti ključni za razumevanje osnovne strukture haotičnih sistema.

U celini, teorija haosa pruža uvide u ponašanje kompleksnih sistema i ima primene u različitim oblastima, uključujući fiziku, biologiju, ekonomiju, meteorologiju i računarstvo. Ona nam pomaže da razumemo fenomene kao što su turbulencija, dinamika populacije, vremenski obrasci i ponašanje finansijskih tržišta.

Mapiranje

U kontekstu steganografije slika kod chaos-based algoritma, mapiranje se odnosi na proces povezivanja haotičnog signala (generisanog dinamičkim sistemom) sa određenim delovima slike. Ovaj proces mapiranja se obično odnosi na pridruživanje vrednosti haotičnog signala sa pikselima slike ili nekim drugim elementima slike.

Kada kažemo "mapiranje", zapravo označavamo vezivanje vrednosti iz haotičnog signala sa odgovarajućim delovima slike koji će se koristiti za skrivanje tajnih informacija. Na primer, svaka vrednost haotičnog signala može biti mapirana na intenzitet boje određenog piksela, ili na poziciju određenih tekstura ili oblika unutar slike.

Ovim mapiranjem se stvara veza između haotičnog signala i slike, omogućavajući tajnim informacijama da budu skrivene unutar slike na način koji je teško otkriti ili dekodirati bez odgovarajućih ključeva i parametara. Mapiranje je ključni korak u procesu steganografije slika koji omogućava efikasno i sigurno skrivanje tajnih informacija.

Haotična mapiranja su matematički modeli koji prikazuju kompleksno i nepredvidivo ponašanje sistema kroz iterativno ponavljanje određene funkcije. Dva poznata primera haotičnih mapa su Logističko mapiranje i Henonovo mapiranje, koji se koriste u različitim naučnim disciplinama zbog svojih interesantnih karakteristika i sposobnosti da generišu haotične dinamičke obrasce. Pored njih koriste se i Chuaov oscilator, Rosslerov oscilator, Lorenzov sistem i sinhronizacija haotičnih mapa.

Logističko mapiranje

- **Formula**:

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n)$$

, gde x_n predstavlja stanje sistema u trenutku *n*, a *r* je parametar koji kontroliše dinamiku sistema.

- **Karakteristike**:

Logističko mapiranje prikazuje različite dinamičke obrasce u zavisnosti od vrednosti parametra r. Za neke vrednosti r sistem konvergira ka stabilnoj tački, dok za druge vrednosti izražava periodične oscilacije ili haotično ponašanje.

Postoji kritična vrednost parametra r, nazvana tačka bifurkacije r_c, iznad koje sistem ulazi u haos. Ovaj fenomen je poznat kao periodična bifurkacija.

Logističko mapiranje pokazuje iterativno bifurkaciono ponašanje pri povećanju vrednosti parametra *r*, što dovodi do kompleksne strukture pod nazivom bifurkacioni dijagram.

Henonovo mapiranje

- **Formule**:

$$x_{n+1} = y_n + 1 - a \cdot x_n^2$$

$$y_{n+1} = b \cdot x_n$$

gde x_n i y_n predstavljaju koordinate tačke u faznom prostoru, a a i b su parametri koji kontrolišu dinamiku sistema.

- **Karakteristike**:

Henonovo mapiranje je jednostavna, ali moćna iterativna funkcija koja generiše haotične dinamičke obrasce u dvodimenzionalnom faznom prostoru. Mapiranje se koristi za generisanje fraktalnih struktura, a takođe se koristi u kriptografiji i obradi signala.

Henonovo mapiranje karakteriše prisustvo haotičnih atraktora u faznom prostoru, što ga čini korisnim alatom za modeliranje složenih dinamičkih sistema.

Chuaov oscilator

- **Karakteristike**

Chuaov oscilator je dinamički sistem koji pokazuje kompleksno i haotično ponašanje. Iako je prvobitno predstavljen u kontekstu električnih kola, Chuaov oscilator se može koristiti kao osnova za razvoj haos algoritama u steganografiji U elektrotehnici Chuaov oscilator sastoji se od nelinearne otpornosti, kapaciteta i induktivnosti, zajedno sa aktivnim komponentama kao što su operacioni pojačavači.

Parametri Chuaovog oscilatora mogu se koristiti kao osnova za generisanje ključeva. Različite vrednosti parametara mogu generisati različite haotične obrasce, koji se zatim koriste za enkripciju ili generisanje ključa. Haotični signal može se koristiti za određivanje lokacija unutar medijuma u koje će se umetnuti tajna poruka. Ovaj proces se može izvesti na način koji minimizira uticaj na vizuelni izgled ili kvalitet medijuma. Primalac poruke koristi iste parametre Chuaovog oscilatora kako bi rekonstruisao haotični signal i izvukao skrivene informacije iz medijuma.

Rosslerov oscilator

- **Karakteristike**

Rosslerov oscilator je još jedan dinamički sistem koji pokazuje haotično ponašanje i može se koristiti u haos algoritmima steganografije. Prvi korak je generisanje haotičnog ključa pomoću Rosslerovog oscilatora. Rosslerov oscilator proizvodi trodimenzionalne haotične oscilacije, čije se vrednosti mogu koristiti kao ključ za enkripciju tajnih informacija. Korišćenje Rosslerovog oscilatora u haos algoritmu steganografije omogućava sigurno skrivanje tajnih informacija unutar slike na način koji je teško otkriti ili dekodirati bez odgovarajućih ključeva i parametara.

Lorenzov sisetem

- **Karakteristike**

Lorenzov sistem je dinamički sistem koji se koristi u proučavanju atmosferske konvekcije i koji pokazuje kompleksno haotično ponašanje. Koristi se kao i Rosslerov algoritam. Lorenzov sistem proizvodi trodimenzionalne haotične oscilacije, čije se vrednosti mogu koristiti kao ključ za enkripciju tajnih informacija. Primalac slike koristi iste parametre Lorenzovog sistema kako bi rekonstruisao haotični signal i izvukao tajne informacije iz slike.

Sinhronizacija haotičnih mapa

Sinhronizacija haotičnih mapa u steganografiji predstavlja metodološki pristup koji koristi više haotičnih sistema ili mapa koje su međusobno povezane kako bi se obezbedilo sigurno i efikasno skrivanje tajnih informacija. Ova tehnika može biti korisna u steganografskim aplikacijama gde se zahteva dodatni nivo sigurnosti ili složenosti.

Korišćenje haotičnih mapa

Korišćenje haotičnih mapa u steganografiji podrazumeva generisanje haotičnog signala pomoću odabrane haotične mape, mapiranje tajnih informacija na taj haotični signal, umetanje tih informacija u pokrivajući medijum poput slike ili audio datoteke, i na kraju, kodiranje tih informacija na način koji je teško otkriti bez odgovarajućih ključeva.

U ovom kontekstu, haotične mape se koriste kao sredstvo za stvaranje nosača tajnih informacija koji je teško predvideti ili otkriti bez znanja odgovarajućih parametara. Ovaj pristup omogućava pouzdan i siguran način za skrivanje tajnih informacija u digitalnim medijima, čime se osigurava privatnost i bezbednost komunikacije u digitalnom svetu.

Koraci prilikom kreiranja algoritma i mapiranaj su sledeći:

1. **Izbor Haotične Mape:** Prvi korak je odabir odgovarajuće haotične mape. Česti izbori uključuju Logističko Mapiranje, Henonovo Mapiranje, Lorenzov Sistem, Čuov oscilator i druge, koje smo obradili u prethodnom poglavlju. Svaka mapa ima jedinstvene karakteristike koje mogu uticati na proces umetanja.

2. **Generisanje Haotičnog Signala**: Haotična mapa se iterativno izvršava sa početnim uslovima ili parametrima kako bi se generisao haotični signal. Ovaj signal služi kao nosilac za sakrivanje tajnih podataka. Haotični signal pokazuje osetljivu zavisnost o početnim uslovima, što znači da male promene u početnim parametrima dovode do drastično različitih izlaza, osiguravajući sigurnost sakrivenih informacija.

3. **Mapiranje Tajnih Podataka**: Tajni podaci se obično pretvaraju u binarni niz ili tok bitova. Svaki bit tajne poruke zatim se mapira na haotični signal. Ovaj proces mapiranja može se postići kroz različite tehnike, poput korišćenja haotičnog signala za određivanje pozicija piksela u slici gde će se podaci umetnuti.

4. **Umetanje Procesa**: Kada je mapiranje završeno, tajni podaci se umeću u pokrivajući medijum. U slučaju slika, ovo često uključuje modifikaciju najmanje značajnih bitova (LSB) vrednosti piksela kako bi se kodirala tajna poruka. Alternativno, u audio steganografiji, amplituda ili faza audio uzoraka može se manipulisati.
5. **Enkripcija (Opciono)**: Da bi se poboljšala sigurnost, tehnike enkripcije se mogu primeniti na tajnu poruku pre njenog umetanja. Haotične mape takođe se mogu

koristiti u procesu enkripcije za generisanje ključeva ili za mešanje tajnih podataka.

6. **Povratak i Dekodiranje**: Da bi povratili sakrivene informacije, primalac mora posedovati istu haotičnu mapu i početne parametre koje su korišćene tokom umetanja. Iterativnom primenom haotične mape sa tim parametrima, primalac može rekonstruisati haotični signal. Zatim, kroz proces obrnutog mapiranja, ugrađena poruka se može izdvojiti iz rekonstruisanog signala.

7. **Analiza Sigurnost**i: Sigurnost haotične steganografije se oslanja na težinu predviđanja haotičnog signala bez poznavanja početnih parametara. Jačina enkripcije, nasumičnost haotičnog signala i neprimetnost modifikacija na pokrivajućem medijumu su faktori koji doprinose sigurnosti sakrivenih informacija.

Implementacija Chaos-based algoritma u programskom jeziku Python

Za implemenataciju ovog algoritma korišćen je programski jezik Python. Sam projekat odrađen je u okruženju PyCharm, koje nam olakšava ceo proces kodiranja ovog koda. Korišćene su i biblioteke numpy i PIL (Python Imaging Library). Takođe, radi demonstracije korišćeno je logičko mapiranje.

Enkodiranje

```python
import numpy as np
from PIL import Image
def logistic_map(x, r):
    return r * x * (1 - x)

def generate_chaotic_sequence(length, seed, r=3.99):
    sequence = [seed]
    for _ in range(1, length):
        sequence.append(logistic_map(sequence[-1], r))
    return sequence

def embed_message(image_path, message, seed):
    binary_message = ''.join(format(ord(char), '08b') for char in message)
    message_length = len(binary_message)

    img = Image.open(image_path)
    img_array = np.array(img)
    if img.mode != 'RGB':
        raise ValueError("Image mode needs to be RGB")

    chaotic_sequence = generate_chaotic_sequence(message_length, seed)

```

Slika 1.a

```
22
23      flat_indices = (np.array(chaotic_sequence) * img_array.size // 3).astype(np.int64)
24
25      for idx, bit in zip(flat_indices, binary_message):
26          row = (idx // 3) // img_array.shape[1]
27          col = (idx // 3) % img_array.shape[1]
28          channel = idx % 3
29
30          if bit == '1':
31              img_array[row, col, channel] |= 1;
32          else:
33              img_array[row, col, channel] &= ~1
34
35      stego_image = Image.fromarray(img_array)
36      stego_image.save('stego_image.png')
37
38  embed_message( image_path: 'download.png',  message: 'porukamojajeova',  seed: 0.5)
```

Slika 1.b

Slike 1.a i 1.b predstavljaju komplentu implementaciju alogoritma za enkripciju.

```
import numpy as np
from PIL import Image
```

Slika 2.

Na slici 2. možemo videti uvoz neophofnih biblioteka. *import numpy as np* je uvoz biblioteke NumPy koja pruža podršku za rad sa višedimenzionalnim nizovima i matricama. *from PIL import Image* je zapravo uvoz klase Image iz biblioteke PIL (Python Imaging Library) koja omogućava manipulaciju slikama.

```
def logistic_map(x, r):
    return r * x * (1 - x)
```

Slika 3.

logistic_map(x, r) funkcija predstavlja logističko mapiranje. Ova funkcija uzima vrednosti *x* i *r* kao argumente i računa sledeću vrednost logističkog mapiranja.

Vrednost parametra *x* predstavlja stanje sistema u trenutku *n*, dok *r je* parametar koji kontroliše dinamiku sistema.

```
def generate_chaotic_sequence(length, seed, r=3.99):
    sequence = [seed]
    for _ in range(1, length):
        sequence.append(logistic_map(sequence[-1], r))
    return sequence
```

Slika 4.

generate_chaotic_sequence(length, seed, r=3.99) je funkcija koja generiše haotični niz koristeći logističko mapiranje. Dužina niza se zadaje parametrom *length*, početna vrednost (seme) se zadaje parametrom *seed*, a parametar *r* predstavlja parametar logističkog mapiranja (podrazumevana vrednost je 3.99).

```
13  def embed_message(image_path, message, seed):
14      binary_message = ''.join(format(ord(char), '08b') for char in message)
15      message_length = len(binary_message)
16
17      img = Image.open(image_path)
18      img_array = np.array(img)
19      if img.mode != 'RGB':
20          raise ValueError("Image mode needs to be RGB")
21
22      chaotic_sequence = generate_chaotic_sequence(message_length, seed)
23
24      flat_indices = (np.array(chaotic_sequence) * img_array.size // 3).astype(np.int64)
25
26      for idx, bit in zip(flat_indices, binary_message):
27          row = (idx // 3) // img_array.shape[1]
28          col = (idx // 3) % img_array.shape[1]
29          channel = idx % 3
30
31          if bit == '1':
32              img_array[row, col, channel] |= 1;
33          else:
34              img_array[row, col, channel] &= ~1
35
36      stego_image = Image.fromarray(img_array)
37      stego_image.save('stego_image.png')
```

Slika 5.

Funkcija embed_message(image_path, message, seed)`: je glavna funkcija koja vrši umetanje tajne poruke u sliku. Ova funkcija uzima putanju do slike *image_path*, tajnu poruku *message* koja se skriva, i *seed* koji se koristi za generisanje haotičnog niza.

Linija koda 14 (b inary_message = ".join(format(ord(char), '08b') for char in message)) konvertuje svaki karakter tajne poruke u binarni oblik. Svaki karakter se prvo konvertuje u ASCII kod, a zatim se formatira kao 8-bitni binarni broj. Nakon toga se izvlači duzina same poruke koristeći *len* funkciju koja kao parametar prima string za koji želimo da vratimo dužinu.

Nakon toga se u liniji 17, uz pomoć PIL biblioteke, učitava slika u koju želimo da upisemo nasu tajnu poruku. Slika se dalje konvertuje u niz radi lakše manipulacije u liniji 18 uz pomoć NumPy biblioteke. Veoma važna stvar je da slika mora biti u RGB režimu, ukoliko nije potrebno je baciti izuzetak što mi i radimo u linijama 19 i 20.

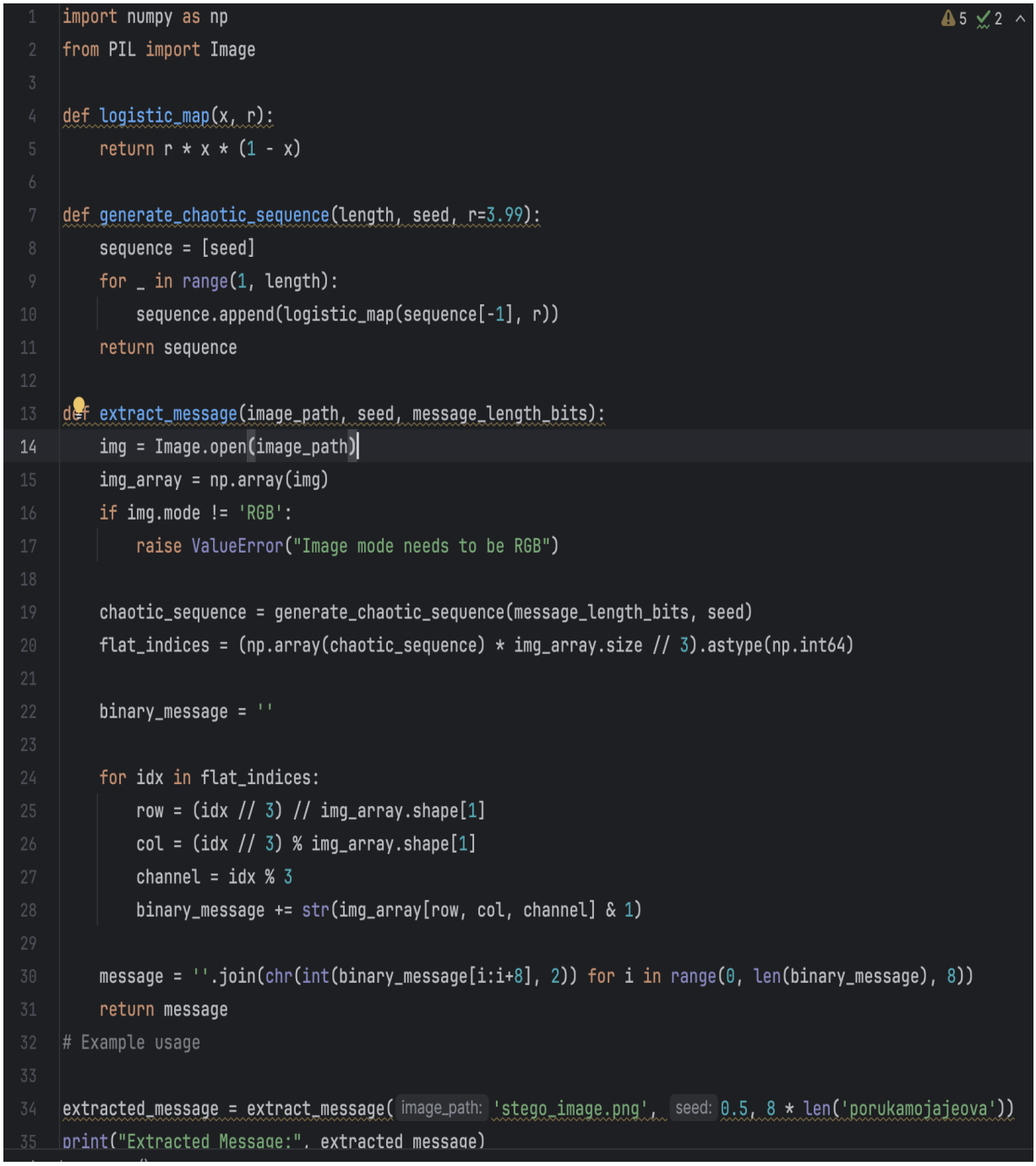
Kada to zavrsimo, gotova su sva pripremanja naše slike u koju želimo da upišemo tajnu poruku. Generisanje haotičnog niza se vrši pomoću prethodno definisane i objašnjene funkcije generate_chaotic_sequence. Nakon generisanja haotičnog niza, potrebno je izračunava indekse piksela u ravnom nizu slike na osnovu haotičnog niza. Ovo izračunavanje se vrši u liniji 24. *img_array.size // 3 r*ačuna broj piksela u slikama sa tri kanala (RGB), a *astype(np.int64)* pretvara indekse u celobrojni tip.

Nakon toga potrebno je iterirati kroz parove indeksa piksela i bitova tajne poruke. Ovo je najlakse odraditi funkcijom *zip*, koju i koristimo u liniji 26. U petlji, za svaki par indeksa piksela i bita tajne poruke, određuje red, kolonu i kanal piksela. Ako je bit 1, postavlja najmanji bit piksela na 1. Ako je bit 0, postavlja najmanji bit piksela na 0. Na kraju, čuva rezultujuću sliku kao "stego_image.png".

Slika 6.a Pre *Slika 6.b Posle*

Slika 6.a predstavlja sliku u kojoj krijemo našu tajnu poruku, dok slika 6.b predstavlja rezultujuću sliku nakon primene Chaos based algoritma sa logičkim mapiranjem. Ovo što je bitno napomenuti je da je kvalitet slike ostao isti i finese dodavanja poruke nisu primetne.

Dekripcija



```python
import numpy as np
from PIL import Image

def logistic_map(x, r):
    return r * x * (1 - x)

def generate_chaotic_sequence(length, seed, r=3.99):
    sequence = [seed]
    for _ in range(1, length):
        sequence.append(logistic_map(sequence[-1], r))
    return sequence

def extract_message(image_path, seed, message_length_bits):
    img = Image.open(image_path)
    img_array = np.array(img)
    if img.mode != 'RGB':
        raise ValueError("Image mode needs to be RGB")

    chaotic_sequence = generate_chaotic_sequence(message_length_bits, seed)
    flat_indices = (np.array(chaotic_sequence) * img_array.size // 3).astype(np.int64)

    binary_message = ''

    for idx in flat_indices:
        row = (idx // 3) // img_array.shape[1]
        col = (idx // 3) % img_array.shape[1]
        channel = idx % 3
        binary_message += str(img_array[row, col, channel] & 1)

    message = ''.join(chr(int(binary_message[i:i+8], 2)) for i in range(0, len(binary_message), 8))
    return message
# Example usage

extracted_message = extract_message( image_path: 'stego_image.png', seed: 0.5, 8 * len('porukamojajeova'))
print("Extracted Message:", extracted_message)
```

Slika 7.

Na početku kao i kod enkripcije imamo uvoz biblioteka NumPy i PIL. Takođe, implementirane su logistic_map(x, r) i generate_chaotic_sequence(length, seed, r=3.99). Ove dve funkcije su identične kao kod enkripcije. Da bi se uspešno dekriptovala i izvukla poruka potrebno je da se znaju početni parametri da bi se generisali iste vrednosti za lokacije čuvanja poruka.

extract_message(image_path, seed, message_length_bits) je glavna funkcija koja ekstrakuje tajnu poruku iz slike. Funkcija uzima putanju do slike *image_path*, početno seme *seed* i dužinu poruke u bitovima *message_length_bits*.

Isto kao u enkripciji, potrebno je otvoriti i ucitati sliku, zatim je konvertovati u niz bitova. Nakon toga ide provera moda u kome je slika, takođe je bitno da bude u *RGB* modu i ukoliko nije baca se izuzetak.

Da bi izvukli sliku potrebno je kreirati istu sekvencu haos signala da bi smo mogli da vidimo na kojim lokacijama u slikama se nalazi zapravo naša poruka. Linija 24 računa indekse nizova slike gde se nalazi poruka. U petlji se izračunava red, kolona i kanal piksela, a zatim se najmanji bit kanala koristi za rekonstrukciju binarne poruke. Na kraju, binarna poruka se dekodira u originalnu tekstualnu formu.

Na kraju vidimo primer korišćenja funkcije za izvlačenje poruke gde se ekstrakuje skrivena poruka iz slike *stego_image.png* koristeći početno seme 0.5 i dužinu poruke koja je prethodno enkodirana. Slika *stego_image.png* se dobija uz pomoć enkripcije određene slike gde je poruka sakrivena. I poslednje je ispisivanje poruke na standardnom izlazu.

Sigurnosti sistema

Haosom zasnovana steganografija slika integriše nepredvidivu i izuzetno osetljivu prirodu haotičnih sistema sa umetnošću skrivanja informacija unutar digitalnih slika. Ova metoda nudi sofisticiran pristup steganografiji, pružajući jedinstvenu kombinaciju sigurnosnih karakteristika koje koriste matematičku složenost teorije haosa. U nastavku je analiza sigurnosnih karakteristika koje ova tehnika pruža, kao i o potencijalnim napadima i protivmerama, sa posebnim osvrtom na dobre i loše strane.

Zaštitni znak haotičnih sistema je njihova ekstremna osetljivost na početne uslove. Ova osobina se koristi u haosom zasnovanoj steganografiji za generisanje šablona ugrađivanja koji su visoko zavisni od tajnih ključeva (početnih uslova). Čak i mala promena u ključu rezultira dramatično drugačijim šablonom ugrađivanja, čineći neovlašćeno otkrivanje i izvlačenje skrivene poruke izuzetno teškim bez tačnog znanja o početnim uslovima.

Pored toga, nutrašnja nepredvidljivost haotičnih mapa osigurava da šablon ugrađenih informacija ne pokazuje prepoznatljive regularnosti. Ova slučajnost otežava statističkim metodama steganalize da razlikuju između originalnih slika i stego-slika (slika sa skrivenim podacima).

Složene dinamike haotičnih sistema mogu se iskoristiti za kreiranje algoritama ugrađivanja koji su otporni na manje modifikacije, kao što su kompresija i šum, osiguravajući integritet skrivene poruke čak i kada stego-slika prolazi kroz uobičajene forme digitalne manipulacije.

Losa strana, proces generisanja haotičnih sekvenci i mapiranja na piksele slike može biti računarski zahtevan, posebno za velike slike ili poruke. To može ograničiti praktičnost haosom zasnovanih metoda u scenarijima gde su računarski resursi ograničeni ili gde je kritična realno-vremenska izvedba.

Sigurnost i efikasnost steganografske metode visoko zavise od izbora parametara haotične mape i početne vrednosti semena. Loše odabrani parametri mogu dovesti do manje efikasnih šablona ugrađivanja ili smanjene osetljivosti, potencijalno kompromitujući sigurnost.

Ako napadači imaju pristup parovima originalnih i stego-slika, mogli bi potencijalno analizirati ove parove kako bi dedukovali haotični šablon ili reverse-engineer-ovali algoritam ugrađivanja, predstavljajući pretnju poverljivosti skrivene poruke.

Potencijalni napadi i protivmere

Napadači bi mogli koristiti napredne statističke metode za analizu distribucije piksela slika, ciljajući da otkriju anomalije uvedene procesom ugrađivanja. Ovaj tip napada teži identifikaciji prisustva skrivene poruke, a ne njenoj ekstrakciji.

Napadi Poznatog Teksta: Kao što je pomenuto, pristup i originalnim i modifikovanim slikama mogao bi omogućiti napadačima da izvrše diferencijalne analize, potencijalno otkrivajući tajne steganografskog algoritma.

Napadi Grubom Silom: Teorijski, napadač bi mogao pokušati da napadne tajni ključ koji se koristi za generisanje haotične sekvence. Međutim, s obzirom na visoku osetljivost haotičnih sistema na početne uslove, prostor mogućih ključeva je ogroman, čineći napade grubom silom nepraktičnim sa trenutnim računarskim sposobnostima.

Postoje nekoliko načina da se spreče ovi napadi:

Poboljšana Sigurnost Ključa: Implementacija robustnih praksi upravljanja ključevima, kao što su korišćenje velikih, nasumičnih ključeva i njihova redovna promena, može značajno ublažiti rizik od napada grubom silom i napada poznatog teksta.

Varijabilnost Parametara: Varijabilnost parametara haotične mape i algoritma ugrađivanja za različite poruke ili različite delove iste poruke može poboljšati sigurnost, otežavajući napadačima predviđanje ili reverse-engineering sistema.

Hibridni Pristupi: Kombinovanje haosom zasnovane steganografije sa kriptografskim tehnikama može dodati dodatni sloj sigurnosti. Šifrovanje poruke pre njenog ugrađivanja koristeći haotični šablon može zaštititi od neovlašćenog izvlačenja čak i ako je prisustvo poruke otkriveno.

Zaključno, dok haosom zasnovana steganografija slika nudi obećavajuće sigurnosne karakteristike koristeći složenost i nepredvidljivost haotičnih sistema, nije bez svojih izazova. Balansiranje prednosti protiv slabosti i implementacija protivmera protiv potencijalnih napada ključni su za poboljšanje sigurnosti i efikasnosti ove steganografske metode.

Zaključak

Steganografija sa haos baziranim algoritmom predstavlja moćan i efikasan metod za sigurno skrivanje informacija unutar digitalnih medija. Korišćenje haotičnih sistema pruža visok nivo sigurnosti i pouzdanosti, što čini ovu metodu izuzetno korisnom u mnogim aplikacijama gde je privatnost ključna.

Korišćenje haotičnih mapa u steganografiji omogućava efikasno skrivanje informacija unutar digitalnih medija. Haotične mape generišu kompleksne signale koji se mogu koristiti za kodiranje tajnih poruka, čineći ih teško detektovanim. Haotični algoritmi obezbeđuju visok nivo sigurnosti u steganografskim aplikacijama. Zbog nepredvidljivog ponašanja haotičnih sistema, teško je otkriti ili dekodirati skrivene poruke bez tačnih parametara ili ključeva. Korišćenje haotičnih mapa omogućava skrivanje informacija bez detekcije od strane napadača ili nepoželjnih posmatrača. Tajne poruke se mogu sigurno prenositi kroz javne kanale bez straha od otkrivanja.

Haotični algoritmi pružaju raznovrsne mogućnosti za steganografsko skrivanje informacija. Različite haotične mape i parametri mogu se koristiti za prilagođavanje specifičnim zahtevima aplikacije i pružanju dodatnih slojeva sigurnosti. Iako haotični algoritmi pružaju visok nivo sigurnosti, neophodno je pažljivo upravljati ključevima ili parametrima kako bi se osiguralo da samo autorizovane strane mogu pristupiti skrivenim informacijama.

Reference

[1] *Kahn, D. (1996). The history of steganography. In: Anderson, R. (eds) Information Hiding. IH 1996. Lecture Notes in Computer Science, vol 1174. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-61996-8_27*

[2] *https://www.sciencedirect.com/science/article/abs/pii/S0165168409003648?via%3Dihub*

[3] *https://www.sciencedirect.com/science/article/abs/pii/S0030402620313280*